Recursive analytic spherical harmonics gradient for spherical lights

Pierre Mézières, Nicolas Mellado, Loic Barthe, Mathias Paulin

▶ **To cite this version:**

Pierre Mézières, Nicolas Mellado, Loic Barthe, Mathias Paulin. Recursive analytic spherical harmonics gradient for spherical lights. Computer Graphics Forum, 2022, 41 (2), pp.1-14. 10.1111/cgf.14482 . hal-03622389

HAL Id: hal-03622389
https://hal.science/hal-03622389

Submitted on 28 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

EUROGRAPHICS 2022 / R. Chaine and M. H. Kim
(Guest Editors)

Volume 41 (2022), Number 2

Recursive analytic spherical harmonics gradient for spherical lights

P. Mézières, N. Mellado, L. Barthe, M. Paulin

IRIT, Université de Toulouse, CNRS, INP, UT3, Toulouse, France

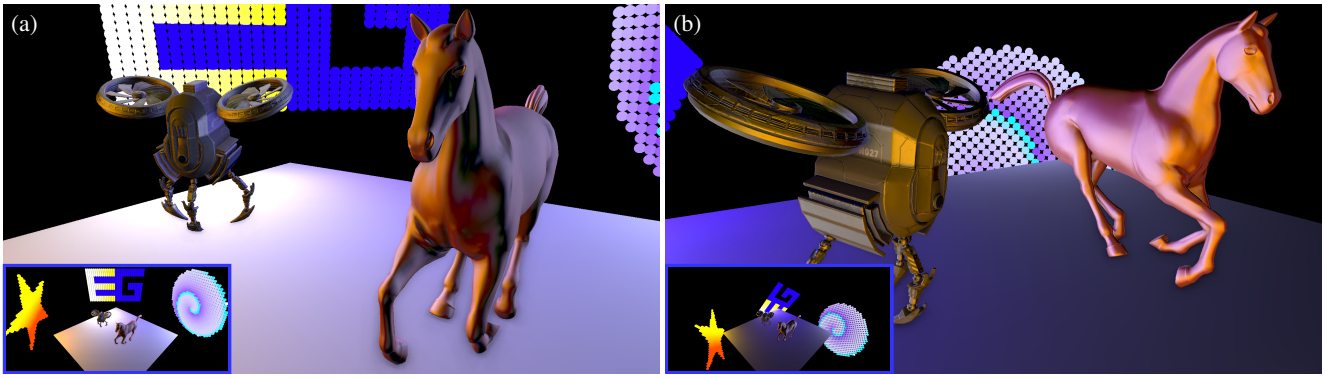


Figure 1: *This fully dynamic scene is rendered at 77 fps with 1793 lights in an image of resolution* 1920×1080*. This is achieved by introducing new analytic formulae that efficiently computes the spherical harmonics (SH) gradients for lighting with uniform spherical lights. We integrate our new computations in a SH based rendering system able to compute the lighting from several hundreds of light sources in real time. This rendering system first computes the SH coefficients and their gradients on a 3D grid, and then compute the shading at each visible point by interpolating the samples grid. Our formulae reduce the gradients computational cost and allow to: increase the frame rate, increase the number of SH bands, use more lights and/or increase the 3D grid resolution.*

Abstract

When rendering images using Spherical Harmonics (SH), the projection of a spherical function on the SH basis remains a computational challenge both for high-frequency functions and for emission functions from complex light sources. Recent works investigate efficient SH projection of the light field coming from polygonal and spherical lights. To further reduce the rendering time, instead of computing the SH coefficients at each vertex of a mesh or at each fragment on an image, it has been shown, for polygonal area light, that computing both the SH coefficients and their spatial gradients on a grid covering the scene allows the efficient and accurate interpolation of these coefficients at each shaded point. In this paper, we develop analytical recursive formulae to compute the spatial gradients of SH coefficients for spherical light. This requires the efficient computation of the spatial gradients of the SH basis function that we also derive. Compared to existing method for polygonal light, our method is faster, requires less memory and scales better with respect to the SH band limit. We also show how to approximate polygonal lights using spherical lights to benefit from our derivations. To demonstrate the effectiveness of our proposal, we integrate our algorithm in a shading system able to render fully dynamic scenes with several hundreds of spherical lights in real time.

CCS Concepts

• ***Computing methodologies*** → ***Rendering; Rasterization;*** *Real-time simulation;*

1. Introduction

Solving the rendering equation, both for real-time and offline applications, relies on the convolution of spherical functions representing the appearance of an object, the BSDF, and the incident light field from the light sources. Whereas Spherical Harmonics (SH) are a useful representation of spherical functions, projecting an arbitrary function on the SH basis is expensive to compute, which limits the number of projected functions per frame or/and adds strong limitations on their reconstructed frequencies.

Solving this computational issue of SH to enable faster and more

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

accurate rendering of scenes lit by several hundreds of area lights is thus an important challenge. Ideally, the SH coefficients of the incident light field should be computed at each visible point or at each vertex of the scene. However, as noted by Wu *et al.* [WCZR20], the light field coming from area lights is spatially smooth, as well as its SH coefficients, and this per-point computation can be efficiently performed by interpolating SH coefficients stored in a volumetric grid. Computing and storing spatial gradients of the SH coefficients on such a grid then enables a smooth and accurate interpolation. Wu *et al.* [WCZR20] demonstrate the significant gain in rendering time provided by this approach by introducing the computation of the SH gradients for polygonal lights in a Precomputed Radiance Transfer (PRT) framework [SKS02].

In this paper, rather than considering polygonal lights, we derive a SH coefficient gradient formulation for spherical lights. Our new formulation is very efficient (Fig. 1). Based on this formulation, we propose a method to approximate polygonal lights by spherical ones. We thus end-up with a faster gradient computation, applicable to a large variety of light shapes.

This is achieved by introducing three new contributions:

- an efficient computation of the spatial gradient of SH coefficients for spherical lights (Sec. 4),
- an efficient computation of spatial gradient of SH basis functions (Sec. 4.3),
- a model approximating polygonal lights SH projection using spherical lights (Sec. 4.4).

These contributions are then integrated into a real-time rendering framework that both computes SH coefficients and their gradients on the vertices of a volumetric grid, and interpolates the SH coefficients at each visible fragment using gradient-based methods (as first-order Taylor or Hermite interpolation).

In the following, we express the gradient of SH coefficients (Sec. 4.2) by differentiating the analytical projection of the light field emitted by spherical lights onto the SH basis proposed in [MDVP22]. Due to differentiation chain rule, we need to compute the spatial gradient of the SH basis functions (Sec. 4.3). We derive our gradient formulation using the principle followed by Sloan [Slo13] so that their computation is fast. Even though our formulations are specifically derived for spherical lights, they can trivially be extended to any circularly symmetric function, and the computation of the gradient of SH basis functions is general.

We evaluate the scalability of our method w.r.t the number of lights and the interpolation grid resolution in Sec. 6, after showing how it has been integrated in a recent framework for dynamic SH lighting [MP21] in Sec. 5.

2. Related works

Spherical Harmonics lighting Real Spherical Harmonics is an orthogonal function basis well suited to solve the rendering equation [Kaj86]. It was first introduced in Computer Graphics for computing lighting from an environment map [CMS87]. SH provide a frequency decomposition of spherical functions whose precomputation and storage is very efficient for low frequency approximation, even though remaining more expensive when representing high frequencies. Taking advantage of this property, SH are successfully applied in Precomputed Radiance Transfer (PRT) [SKS02]. In contrast, wavelet decomposition offers an all-frequency approach [NRH03] but wavelet coefficients are more expensive to compute and store.

The usefulness of SH decomposition for PRT has led to several contributions, *e.g.* for extending PRT to dynamic and deformable scenes [SLS05], for visibility and shadow approximation [RWS*06], or for fast projection using Zonal Harmonics subbasis [NSF12, BXH*18]. We refer readers interested by PRT to the surveys by Lehtinen [Leh07] and Ramamoorthi [Ram09].

Area lights Area lights are far more convenient to compute physically-based shading than simple light primitives, such as point, directional or spot lights. Nevertheless, their use for real-time rendering is also more complex and expensive. The integration of the lighting function over an area light source was studied for different representations of the light source. While Monte-Carlo integration provides a ground-truth solution for lighting, analytical integration schemes introduce less stochastic error while being specific to the area light representation. Dealing with polygonal representation of the area light, Heitz *et al.* [HDHN16] propose a method called *Linearly Transformed Cosines* (LTC) to approximate the lighting by transforming the projection of the polygon onto the unit sphere toward a clamped cosine according to the BRDF distribution. To improve the lighting computation of area lights, Peters couples LTC with importance sampling of the BRDF [Pet21], and Allmenröder and Peters [AP21] combine LTC with SH. Dealing with spherical representation of the area light, Dupuy *et al.* [DHB17] propose a spherical cap parameterization enabling the use of an analytic integration and an importance sampling scheme of the lighting function.

These methods are however limited to a small number of lights to provide real-time performances and the pre-computation of the radiance transfer is an effective solution to compute lighting with numerous lights. For SH based lighting, some approaches introduce an analytical solution to compute the SH coefficients of polygonal lights [WR18, BXH*18]. Following the approach introduced in Wang and Ramamoorthi [WR18], Wu *et al.* [WCZR20] propose to pre-compute and store the SH coefficients and their spatial gradients on the vertices of a low resolution 3D grid. These coefficients are then interpolated during rendering, which allows to handle several hundreds of polygonal lights in real-time, while remaining limited to static scenes.

Radiance caching Irradiance and radiance caching are widely used to compute global illumination by reconstructing the lighting function from a sparse set of probes caching the global light field [WRC88, KGPB08, KGW*08]. To improve the sparsity of the cache and enhance the interpolation accuracy, a solution is to use the function gradients [WH08, KGBP05, JDZJ08, JZJ08]. These methods estimate the gradient numerically using Monte-Carlo; in contrast, we compute the gradient of the SH-encoded light field analytically. Several recent approaches rely on off line pre-computation of radiance cache for real-time rendering [DSJN19, ZBN19]. Based on a volumetric approach introduced by Greger *et al.* [GSHG98], our method computes radiance values and their gradients on probes at each frame, thus allowing to render fully dynamic lighting.

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

Yet, even using techniques that exploit a sparse set of probes [SL17], the cost of storage becomes high for large and complex scenes. To overcome this limitation, probes data can be compressed with only a low loss in quality [SS21]. Our proposal is orthogonal to these compression methods and can therefore take advantage of their benefits.

Spherical Harmonics gradient The spatial gradients of SH coefficients have not been widely used in Computer Graphics. Annen *et al.* [AKDS04] propose to compute the SH coefficients of the incident illumination with their spatial gradient on a sparse set of samples and then reconstruct the SH coefficients on any shaded point. However, gradients are computed with a numerical integration over the area light, which significantly increases computations when considering higher SH frequency bands. On the other hand, based on the initial results for SH coefficient computation on polygonal light [WR18], Wu *et al.* [WCZR20] derive an analytical solution to compute the spatial gradient of SH coefficients on polygonal lights. Our contributions follow the same principle, but are developed for spherical lights.

Differentiable rendering Computing the gradient of numerous rendering quantities is at the heart of differentiable rendering approaches. Aiming at solving inverse problems [LTL*19, LZBD21, NDDJK21], differentiable rendering frameworks mainly rely on path-space [ZMY*20] or physical formulation of light transport [ZWZ*19].

While our propositions are not directly related to such differentiable approaches, they may serve as building blocks for improving gradient computations of the incident light field. Indeed, several approaches based on differentiable rendering take advantage of SH. For instance, Liu *et al.* [LTL*19] directly compute the gradient of the SH basis function. The use of our method leads to faster computations. Lyu *et al.* [LHL*21] use a spherical representation of the scene to compute differentiable shadow by projecting the spheres on SH. This approach can thus directly benefit from our method. Technically, rather than relying on finite differences or automatic differentiation frameworks, we introduce an analytical expression optimizing the evaluation of the SH coefficient gradients.

3. Background

Our proposal relies on fundamental principles of SH lighting. After recalling the SH definitions and their use in lighting computation, we describe the methods used in our gradient computations to efficiently evaluate SH basis function [Slo13] and project the spherical lights on SH [MDVP22].

3.1. Lighting with Spherical Harmonics

Using spherical coordinates, the real part of an harmonic Y_l^m is expressed as:

$$Y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}K_l^{|m|}\,P_l^{|m|}(\cos\theta)\,\sin(|m|\phi) & m<0 \\ K_l^0\,P_l^0(\cos\theta) & m=0 \\ \sqrt{2}K_l^m\,P_l^m(\cos\theta)\,\cos(m\phi) & m>0 \end{cases}, \tag{1}$$

where P_l^m denotes the associated Legendre Polynomial, and K_l^m is a normalizing factor:

$$K_l^m = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}\,. \tag{2}$$

Successive orders $l \geq 0$ correspond to increasing frequency bands, in which degrees span $-l \leq m \leq l$. The Zonal Harmonics (ZH), defined for $m=0$, *i.e.* Y_l^0, is a sub-basis of SH particularly useful for representing circularly symmetric signals. We denote $\mathbf{S}_l^m$ the coefficients of the vector $\mathbf{S}$, indexed by the order l and the degree m, resulting from the projection of $S(\omega)$ on the SH basis $\{Y_l^m(\omega)\}$. Projection on SH basis (Eq. 3) and reconstruction from SH coefficients (Eq. 4) are performed as:

$$\mathbf{S}_l^m = \int_\Omega S(\omega)\,Y_l^m(\omega)\,\mathrm{d}\omega\,, \tag{3}$$

$$S(\omega) = \sum_{l=0}^{+\infty}\sum_{m=-l}^{l}\mathbf{S}_l^m\,Y_l^m(\omega) \approx \sum_{l=0}^{N}\sum_{m=-l}^{l}\mathbf{S}_l^m\,Y_l^m(\omega)\,. \tag{4}$$

The SH basis being orthogonal, its use in lighting computation has a particular interest. Denoting $\mathbf{L}$ and $\mathbf{F}$ respectively the projection of the incident radiance $L(\omega)$ and of the reflectance $F(\omega)$ on the real spherical harmonics basis, the reflected radiance is defined as [CMS87]:

$$\int_\Omega L(\omega)\,F(\omega)\,\mathrm{d}\omega = \mathbf{L}\cdot\mathbf{F}\,. \tag{5}$$

In practice, SH coefficients are computed and stored up to an order N and the reconstruction becomes an approximation with a fixed maximal frequency. In this case, the integral in Eq.5 reduces to a sum of $(N+1)^2$ terms, allowing for an efficient evaluation of the equation.

3.2. Efficient SH evaluation

The SH gradient formulae we derive in Sec. 4.3 rely on the efficient SH evaluation method proposed by Sloan [Slo13]. The evaluation of SH basis functions requires the evaluation of the Associated Legendre Polynomials (ALP) P_l^m:

$$P_l^m(\cos\theta) = \sin^m\theta\,\frac{d^m}{d\cos^m\theta}(P_l(\cos\theta))\,. \tag{6}$$

Sloan proposes to pair the $\sin^m\theta$ from ALP with $\sin(m\phi)$ and $\cos(m\phi)$ from SH expression (Eq. 1). These pairings lead to computationally simpler recurrence relations:

$$\begin{aligned} U_m &= \sin^m\theta\cos(m\phi) = xU_{m-1} - yV_{m-1} \\ V_m &= \sin^m\theta\sin(m\phi) = xV_{m-1} + yU_{m-1}\,, \end{aligned} \tag{7}$$

where U_m corresponds to the case where $m>0$ and V_m corresponds to $m<0$, $x=\sin\theta\cos\phi$, and $y=\sin\theta\sin\phi$. As a consequence, $Q_l^m = P_l^m(\cos\theta)/\sin^m\theta$ is a polynomial in z, with $z=\cos\theta$, that does not require the explicit computation of $\sin^m\theta$ and is defined

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

by these five recurrence relations:

$$\begin{aligned} Q_m^m &= (1-2m)Q_{m-1}^{m-1} \\ Q_{m+1}^m &= (2m+1)zQ_m^m \\ Q_l^m &= \frac{(2l-1)zQ_{l-1}^m - (l+m-1)Q_{l-2}^m}{l-m} \\ Q_{m+2}^m &= \frac{(2m+3)(2m+1)Q_m^m z^2 - (2m+1)Q_m^m}{2} \\ Q_{m+3}^m &= \frac{zQ_m^m((2m+5)(2m+3)(2m+1)z^2 - 3(4m^2+8m+3))}{6} . \end{aligned} \tag{8}$$

In order to improve the final computation, the normalization factors $\sqrt{2}$ and K_l^m of SH (Eq. 1) should directly be integrated in the recurrence relations of Q.

3.3. Spherical light projection

Mézières *et al.* [MDVP22] propose an analytical SH projection for uniform spherical lights. Projecting spherical light with uniform emission on the unit sphere centered at the shaded point creates a circularly symmetric spherical signal. This property cancels the signal projection in all bands except for $m = 0$ (*i.e.* the ZH). In the remaining of the paper, we adopt the notations of Fig. 2, namely, $\mathbf{x}$ is the position of the shaded point, $\mathbf{y}$ is the position of the spherical light and r is its radius. Hence, the ZH projection of spherical lights is expressed as:

$$\tilde{\mathbf{L}}_l(\mathbf{x}) = \begin{cases} \sqrt{\frac{\pi}{2l+1}}(P_{l-1}(\alpha) - P_{l+1}(\alpha)) & \text{if } l \neq 0 \\ \sqrt{\pi}(1-\alpha) & \text{otherwise,} \end{cases} \tag{9}$$

where $\alpha = \cos(a(\mathbf{x}))$ and $a(\mathbf{x})$ is the half-angle subtended by the spherical light (Fig. 2) defined as:

$$a(\mathbf{x}) = \arcsin\left(\frac{r}{\|\mathbf{y}-\mathbf{x}\|}\right) . \tag{10}$$

Then, the projection is rotated to any coordinate system [SLS05] following:

$$\mathbf{L}_l^m(\mathbf{x}) = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x}) , \tag{11}$$

where $\omega = (\mathbf{y}-\mathbf{x})/\|\mathbf{y}-\mathbf{x}\|$ is the direction of the symmetry axis in the targeted coordinate system.

Our new contributions, detailed in the next section, are the spatial differentiation of Eq.9 and Eq.11. Note that Eq.11 is valid for any circularly symmetric signal.

4. Gradient computation

We now derive the formulae for the efficient computation of the spatial gradients of the SH coefficients for spherical lights (Eq.9 and 11) w.r.t the position of the shaded point (Sec. 4.1). By applying the differentiation chain rule, we differentiate both the ZH projection of spherical lights (Sec. 4.2) and the SH basis functions (Sec. 4.3). We introduce in Sec. 4.4 an adaptation of the SH projection and the gradient computation to efficiently approximate triangular lights by spherical lights.

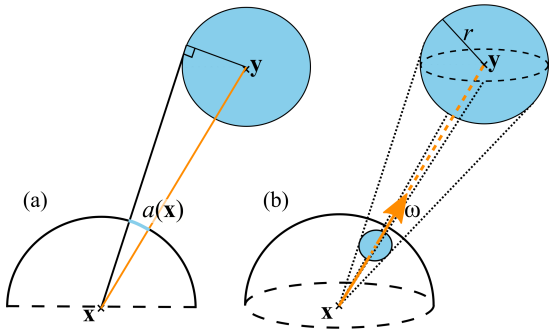


Figure 2: *Notations and geometric setup with a 2D (a) and 3D (b) view where* $\mathbf{x}$ *is a shaded point and* $\mathbf{y}$ *is a spherical light.* $a(\mathbf{x})$ *is the half-angle subtended by the spherical light and* r *is its radius.*

4.1. Overview

We denote $\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ the spatial gradient of the SH coefficients (Eq. 11) w.r.t the shaded point $\mathbf{x}$ (Fig. 2). It is defined as:

$$\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x}) = -\sqrt{\frac{4\pi}{2l+1}}\Big(\underbrace{\nabla_{\mathbf{y}}Y_l^m(\omega)}_{\text{Sec. 4.3}}\overbrace{\tilde{\mathbf{L}}_l(\mathbf{x})}^{\text{Sec. 3.3}} + \overbrace{Y_l^m(\omega)}^{\text{Sec. 3.2}}\underbrace{\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x})}_{\text{Sec. 4.2}}\Big). \tag{12}$$

$\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ represents the gradient according to the translation of a point $\mathbf{x}$. Its computation involves the evaluation of the gradient $\nabla Y_l^m(\omega)$ of the SH basis functions. To compute the latter, it is more convenient to consider that the point $\mathbf{y}$ moves: moving $\mathbf{x}$ along a direction $\mathbf{d}$ is equivalent to move $\mathbf{y}$ along $-\mathbf{d}$. This is what introduces the minus sign and the gradient $\nabla_{\mathbf{y}}$ in Eq.12.

The definition of $\nabla_{\mathbf{x}}\mathbf{L}_l^m(\mathbf{x})$ contains four main terms: the SH functions $Y_l^m(\omega)$, the ZH projection of the spherical light $\tilde{\mathbf{L}}_l(\mathbf{x})$, and their respective gradients $\nabla Y_l^m(\omega)$ and $\nabla\tilde{\mathbf{L}}_l(\mathbf{x})$. In the following, the computation of the gradient $\nabla\tilde{\mathbf{L}}_l(\mathbf{x})$ of ZH coefficients is derived first (Sec. 4.2). Then, based on the SH evaluation method proposed by Sloan, we propose an effective formulation of the gradient $\nabla Y_l^m(\omega)$ of the SH basis functions (Sec. 4.3).

It should be noted that $\tilde{\mathbf{L}}_l(\mathbf{x})$ and $Y_l^m(\omega)$ are two functions that could be written in polynomial Cartesian form for each order and degree [Slo08]. If doing so, computing the spatial gradient would essentially be a matter of differentiating each equations. However, our method is based on a more efficient recursive analytical equations than these polynomial forms (Table 1 and 2).

4.2. Spherical light ZH gradient : $\nabla\tilde{\mathbf{L}}_l(\mathbf{x})$

Our analytical definition of the gradient of the ZH coefficients is derived using the chain rule as follows.

Subtended angle gradient Given Eq.10, the gradient of the half-angle subtended by the spherical light is:

$$\nabla_{\mathbf{y}}a(\mathbf{x}) = \frac{-r}{\|\mathbf{y}-\mathbf{x}\|\sqrt{\|\mathbf{y}-\mathbf{x}\|^2 - r^2}}\frac{\mathbf{y}-\mathbf{x}}{\|\mathbf{y}-\mathbf{x}\|} . \tag{13}$$

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

ZH Gradient Given Eq.9, the gradient of the ZH coefficients corresponding to the projection of a spherical light is:

$$\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x}) = \begin{cases} -\sqrt{\frac{\pi}{2l+1}}\nabla_{\mathbf{y}}a(\mathbf{x})\sin(a(\mathbf{x}))O'_l & \text{if } l \neq 0 \\ \sqrt{\pi}\nabla_{\mathbf{y}}a(\mathbf{x})\sin(a(\mathbf{x})) & \text{otherwise} \end{cases}, \quad (14)$$

where $O'_l = (-P'_{l+1}(\alpha) + P'_{l-1}(\alpha))$. Considering the derivative of the Legendre polynomials:

$$\partial_x P_l(x) = -\frac{(l+1)(xP_l(x) - P_{l+1}(x))}{x^2-1}, \quad (15)$$

we obtain:

$$O'_l = \frac{(l+2)(\alpha P_{l+1}(\alpha) - P_{l+2}(\alpha)) - l(\alpha P_{l-1}(\alpha) - P_l(\alpha))}{\alpha^2-1}. \quad (16)$$

4.3. Spherical harmonics basis function gradient : $\nabla Y_l^m(\omega)$

The computation of the gradient of SH coefficients (Eq. 11) requires the computation of the gradient of SH basis functions. We first perform this computation in spherical coordinates and then make the conversion to Cartesian coordinates.

Partial derivatives w.r.t ϕ The SH partial derivative w.r.t ϕ is :

$$\partial_\phi Y_l^m(\theta,\phi) = \begin{cases} -mY_l^{-m}(\theta,\phi) & m < 0 \\ 0 & m = 0 \\ -mY_l^{-m}(\theta,\phi) & m > 0 \end{cases}, \quad (17)$$

and can be computed without overhead when evaluating the SH.

Partial derivatives w.r.t θ The SH partial derivative w.r.t θ is:

$$\partial_\theta Y_l^m(\theta,\phi) = \begin{cases} \sqrt{2}K_l^{|m|}\,\partial_\theta P_l^{|m|}(\cos\theta)\sin(|m|\phi) & m < 0 \\ K_l^0\,\partial_\theta P_l^0(\cos\theta) & m = 0 \\ \sqrt{2}K_l^m\partial_\theta P_l^m(\cos\theta)\cos(m\phi) & m > 0 \end{cases}. \quad (18)$$

Eq.18 might be computed directly, giving the thereafter called ***Direct*** method. This method improves the one proposed in [LTL*19] using a less expensive formula evaluating the derivatives of the ALP based on the recursion formula given by Hansen [Han88]:

$$\partial_\theta P_l^m(\cos\theta) = \begin{cases} P_l^1(\cos\theta) & m = 0 \\ -\frac{1}{2}\{(l-m+1)(l+m) & \\ P_l^{m-1}(\cos\theta) - P_l^{m+1}(\cos\theta)\} & m > 0 \end{cases}. \quad (19)$$

We however propose a more efficient computation method, called the ***Sloan's like*** method, based on the same principle as the one used for the SH basis function computation (Sec. 3.2). Instead of computing $\partial_\theta P_l^m(\cos\theta)$, we compute $(\partial_\theta P_l^m(\cos\theta))/\sin^m\theta$ to remove $\sin^m\theta$ in other SH terms. When deriving $\partial_\theta P_l^m(\cos\theta)$ and factorizing to have a $\sin^m\theta$ factor, we obtain:

$$\frac{\partial_\theta P_l^m(\cos\theta)}{\sin^m\theta} = \frac{P_l^m(\cos\theta)}{\sin^m\theta}\frac{m\cos\theta}{\sin\theta} + \partial_\theta\left(\frac{P_l^m(\cos\theta)}{\sin^m\theta}\right). \quad (20)$$

The first term of the right member of Eq.20 is exactly Q_l^m (Sec. 3.2) and the last term is $\partial_\theta Q_l^m = T_l^m$. We can thus rewrite Eq.20 as:

$$\frac{\partial_\theta P_l^m(\cos\theta)}{\sin^m\theta} = Q_l^m\frac{m\cos\theta}{\sin\theta} + T_l^m. \quad (21)$$

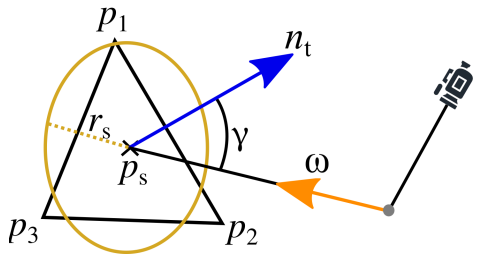


Figure 3: *Approximation of a triangular light with a sphere. The sphere position p_s is the centroid of the triangle (Eq. 24) and the radius r_s is determined so that the area of the yellow disc is equal to the area of the triangle (Eq. 25). γ is the angle between the triangle normal n_t and the direction ω. The shaded point is in grey.*

To compute T_l^m, we derive the recurrence relations of Q_l^m (Eq. 8):

$$\begin{aligned} T_m^m &= 0 \\ T_{m+1}^m &= -(2m+1)(\sin\theta Q_m^m) \\ T_l^m &= \frac{(2l-1)(\cos\theta T_{l-1}^m - \sin\theta Q_{l-1}^m) - (l+m-1)T_{l-2}^m}{l-m} \\ T_{m+2}^m &= -(2m+3)(2m+1)(\sin\theta\cos\theta Q_m^m) \\ T_{m+3}^m &= \frac{(\sin\theta Q_m^m)((-(2m+5)(2m+3)(2m+1)\cos^2\theta)+(4m^2+8m+3))}{2}. \end{aligned} \quad (22)$$

From Eq.22, the final equations integrating the normalization factors of the SH definition are developed in Appendix A.

SH gradient in Cartesian coordinates To obtain the translational gradient in Cartesian coordinates, we need to multiply the gradient computed in spherical coordinates with the Jacobian of the spherical coordinates w.r.t the cartesian coordinates. Since the SH functions vary only in θ and ϕ, we have $\partial_r Y_l^m = 0$, and:

$$\begin{pmatrix} \partial_x Y_l^m \\ \partial_y Y_l^m \\ \partial_z Y_l^m \end{pmatrix} = \begin{pmatrix} \frac{xz}{\Psi} & \frac{-y}{x^2+y^2} \\ \frac{yz}{\Psi} & \frac{x}{x^2+y^2} \\ \frac{-x^2-y^2}{\Psi} & 0 \end{pmatrix} \begin{pmatrix} \partial_\theta Y_l^m \\ \partial_\phi Y_l^m \end{pmatrix}, \quad (23)$$

where $(x,y,z) = \mathbf{y} - \mathbf{x}$ and $\Psi = (x^2+y^2+z^2)\sqrt{x^2+y^2}$.

Validation of our methods We perform several unit tests on CPU (Table 1) to compare the performance of our two methods, ***Direct*** and ***Sloan's like***, with a reference evaluation of the SH functions and their gradients using the polynomial form of SH. The goal is to compare the mean time to evaluate the SH coefficients and their gradient in *one* direction. This comparison uses the same validation procedure than the one proposed by Sloan [Slo13]. For each method, the mean computation time is measured using the same set of thousand directions. For each method, timings are averaged over a hundred executions for each direction. We observe that both the ***Direct*** and the ***Sloan's like*** methods are faster than the polynomial form when having 3 SH bands or more, including the Cartesian conversion. Also, the ***Sloan's like*** method is always faster than the ***Direct*** method.

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

Algorithm		Band limit					
		3	4	5	6	8	10
Polynomial	Eval	27.82	32.22	40.57	54.63	84.51	144.01
	Gradient $\nabla_{x,y,z}$	56.91	83.34	122.28	190.25	386.91	720.26
	Sum	84.74	115.56	162.85	244.87	471.42	864.27
Eval & Gradient $\nabla_{\theta,\phi}$	***Sloan's like*** (ours)	25.86	33.77	44.51	59.95	106.35	169.82
	Direct (ours)	42.28	60.66	81.15	112.32	196.33	296.94
Cartesian conversion		41.63	43.67	48.97	53.82	66.19	84.41
Eval & Gradient $\nabla_{x,y,z}$	***Sloan's like*** (ours)	67.50	77.43	93.48	113.78	172.55	254.23
	Direct (ours)	83.92	104.32	130.13	166.14	262.52	381.35

Table 1: *Timings (in ns, on the CPU) comparison to evaluate SH and their corresponding gradients at different orders. Both our methods are efficient compared to the traditional Cartesian approach. However, our* **Sloan's like** *approach is the most efficient, achieving a speedup of more than ×3 on 10 SH bands. Our gradients expressed w.r.t the spherical direction are converted to obtain the Cartesian spatial gradients (Sec. 4.3). For both our methods, SH are evaluated with Sloan's method [Slo13] and are not separated from the gradient calculations because ∇_ϕ is directly calculated from the SH evaluation (Eq. 17).*

4.4. Approximating polygonal lights SH projection

To take full benefit of our gradient computation for polygonal lights, and as any polygonal light is divisible into triangles, we propose to approximate triangular lights using spherical ones (Fig. 3). We approximate the SH projection of a triangular light by first deducing the spherical light position and radius from the triangle vertices. Then, we modulate the SH projection by a correction factor according to the orientation of the triangle w.r.t the shaded point.

Spherical light parameters The position p_s of the sphere is defined as the centroid of the triangle:

$$p_s = \frac{p_1 + p_2 + p_3}{3}\,, \tag{24}$$

where p_1, p_2 and p_3 are the triangle vertices. The sphere radius is defined so that a disc with the sphere's radius and the triangle have the same area. Reminding that πr^2 measures the area of a disc, and $\|\vec{p_1p_2} \wedge \vec{p_1p_3}\|/2$ the triangle area, the radius of the sphere is:

$$r_s = \sqrt{\frac{1}{2\pi}\|\vec{p_1p_2} \wedge \vec{p_1p_3}\|}\,. \tag{25}$$

Coefficients and gradients For each shading point, the flatness of the triangle is recovered from the spherical shape of the light by multiplying its energy by the factor $|\cos\gamma|$ (Fig. 3), where γ is the angle between the triangle normal n_t and the direction $-\omega$ from the center of the spherical light to the shaded point. As this factor varies for each light at each shaded point, the equation computing the SH coefficients for the light at the shaded point (Eq. 11) needs to be adapted, such that:

$$\mathbf{P}_l^m(\mathbf{x}) = \sqrt{\frac{4\pi}{2l+1}} Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| \tag{26}$$

(we use the absolute value for $|\cos\gamma|$ to consider the triangular light as double sided). This equation multiplies by $|\cos\gamma|$ the SH projection of spherical light proposed by Mézières *et al.* [MDVP22]. Thus, the gradient of this equation is deduced from the gradient of the original equation (Eq. 12) as:

$$\begin{aligned}\nabla_{\mathbf{x}}\mathbf{P}_l^m(\mathbf{x}) = -\sqrt{\frac{4\pi}{2l+1}}\big(&\nabla_{\mathbf{y}} Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| + \\ &Y_l^m(\omega)\nabla_{\mathbf{y}}\tilde{\mathbf{L}}_l(\mathbf{x})\,|\cos\gamma| + Y_l^m(\omega)\tilde{\mathbf{L}}_l(\mathbf{x})\nabla_{\mathbf{y}}\,|\cos\gamma|\big)\,,\end{aligned} \tag{27}$$

with

$$\nabla_{\mathbf{y}}\,|\cos\gamma| = \left(\frac{-n_t}{\|\mathbf{y}-\mathbf{x}\|} + (\mathbf{y}-\mathbf{x})\frac{n_t\cdot(\mathbf{y}-\mathbf{x})}{\|\mathbf{y}-\mathbf{x}\|^3}\right)\frac{\cos\gamma}{|\cos\gamma|}\,. \tag{28}$$

5. Rendering pipeline

The rendering is performed at each frame with two independent passes. The first pass computes SH coefficients and their gradients at each vertex of a 3D grid enclosing the scene. The second pass interpolates the SH coefficients and computes the shading at each visible fragment.

Our rendering pipeline is a clone of the one proposed by Wu *et al.* [WCZR20] in which polygonal lights are replaced by spherical ones. As they do, we can use any of the three interpolations they propose to reconstruct the SH coefficients from the grid values (the linear interpolation, the Taylor interpolation [AKDS04] and the Hermite interpolation [WCZR20]).

The final shading is computed with a recent SH framework handling fully dynamic scenes [MP21]. This method is specialized for separable BRDF. We skip the visibility in our computations, and add a pass of SSAO [BS08] to compensate for the lack of shadows. We also show in (Fig. 11) that our method is versatile and can, for instance, be applied with captured data-driven materials and with per-pixel lighting.

5.1. Efficient parallel grid construction

The method of Wu *et al.* [WCZR20] relies on the ZH factorization [NSF12]. The projection is done on a set of independent ZH

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

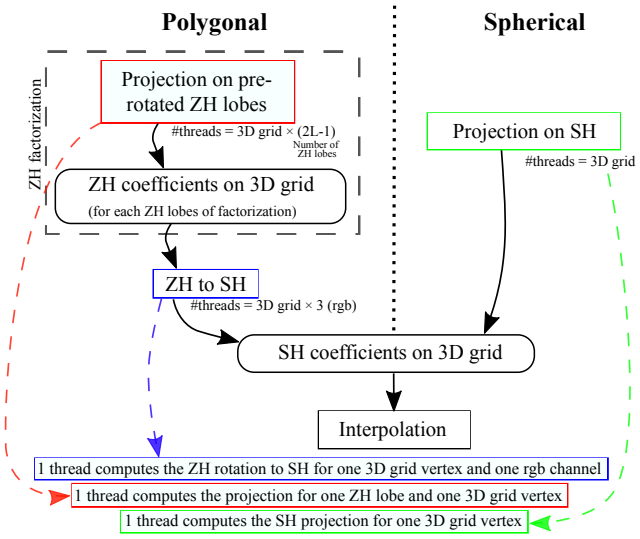
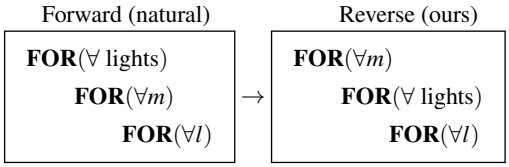


Figure 4: *Parallelization scheme for polygonal (left) and spherical lights (right). The projection of polygonal lights relies on the ZH factorization [NSF12], which is efficiently parallelized and provide a good tradeoff between the number of threads and the register pressure. Our method is factorized following a different model.*

lobes whose directions are fixed for all lights. Each ZH lobe is processed in parallel, which allows a good distribution of the workload (Fig. 4) on the GPU. In our approach, each light defines a ZH lobe direction, which makes the workload difficult to distribute over the different threads, thus increasing the pressure on registers, especially when increasing the SH bands.

To better parallelize the computations, we reverse the main loops in the code. First, as our SH recurrence relations do not use relations between two different degrees m, except for the opposite degree $-m$, we only need to fix a degree m and loop through the orders l. So, the natural approach to compute the SH coefficients and their gradients for all the lights, denoted as *Forward* in the following, is to loop first over the lights, and then over all the degrees m and orders l of SH. As we need to accumulate the SH coefficients for all the lights, this *Forward* strategy needs to keep in GPU registers all the SH coefficients. This requires so many temporary storage that the parallel workload between GPU cores is limited. To improve parallel workload on the GPU, we permute the loops on the lights and on degree m, denoted as *Reverse* in the following. Doing this, we only need to keep in registers the SH coefficients deducted for a given degree m, decreasing the register pressure.



This improves the overall performance of our method, especially when increasing the SH band limit (Table 2). The drawback is that, for each light, we have to recompute all the terms that depend on m. Despite that, the register pressure drop is such that the extra cost of calculation is largely absorbed by a better usage of GPU cores.

SH bands limit	Spherical lights				Polygonal lights	
	Analytical		Ours		Our	Ref.
	forw.	rev.	forw.	rev.	approx.	[WCZR20]
5	4.48	4.32	4.08	2.22	2.53	7.21
6	7.25	6.37	5.98	2.96	3.39	13.08
7	13.11	8.54	7.24	3.77	4.42	16.21
8	19.30	10.41	12.73	4.76	5.35	29.07
9	33.23	12.91	22.26	5.18	5.74	164.01

Table 2: *Timings (in ms) for the computation of the SH and gradients coefficients in a 8^3 3D grid, using 648 lights (spheres or triangles). Columns labeled "analytical" correspond to the SH computations with their Cartesian form, ZH projection is computed in closed form [Slo08] and gradients are computed with analytical derivation. Reverse columns correspond to the inversion of the loops (Eq. 5.1). Our approximation of polygonal lights is computed with the reversed loops and exhibits a low overhead compared to spheres.*

6. Results

We implemented our method in an experimental rendering engine (Sec. 5) using OpenGL / GLSL 4.5 running on an Intel Xeon 2.10 GHz processor and a RTX 2080 NVIDIA graphics card. For our experiments and comparisons, we use the same band limitations as Wu *et al.* did for the polygonal lights [WCZR20]. Lighting is band-limited on 5 SH bands and the convolution with the GGX specular lobes are computed on 9 SH bands. In the following, we denote as "SH reference" the results obtained *without interpolation*. They are obtained by computing the exact projection of the radiance on SH at each pixel instead of only at the 3D grid vertices.

Quality As pointed out in the overview (Sec. 4.1), the SH gradient's coefficients for spherical lights (Eq. 12) can be computed with Cartesian analytic equations for each SH. While being as accurate as these Cartesian analytical forms (Fig. 5), our recursive analytical equations are faster to evaluate (Table 2).

From the three interpolation schemes (trilinear, Taylor and Hermite), the Hermite interpolation is, as demonstrated by Wu *et al.* [WCZR20], the more accurate and the accuracy increases as the grid is denser (Fig. 6).

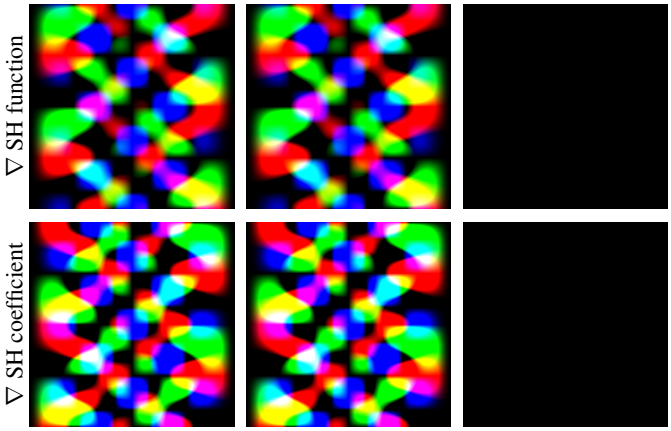
When an equal time is allocated to generate the SH coefficients and their gradients on a 3D grid, or to generate only the coefficients on a 3D grid, the latter grid can be denser. Nevertheless, linear interpolation on such a grid does not reach the quality provided by the Hermite interpolation on a sparser grid (Fig. 7) while using, in this case, 3.35× more memory to store the grid data.

As shown in Fig. 8, reconstructing the SH coefficients with Hermite interpolation from grid samples generates results that closely match the per-pixel direct computation of SH coefficients. We also

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

Scene	Figure	Lights	Output resolution	Eval SH coeff	Eval coeff & grad.	Interp.	ssao	Total	Reference		
									Time	Speed up	MAE ($\times 10^3$)
Drone & Horse	Fig. 1a	1793	1920×1080	3.18	5.30	6.73	0.9	12.93	139.11	10.8×	0.2022
Dragon	Fig. 6	578	1920×1080	0.84	1.82	12.41	0.98	15.21	76.54	5.0×	0.2372
Perseverance	Fig. 7	512	1920×1080	0.86	1.39	10.17	1.06	12.62	55.40	4.4×	0.1310
Armadillo & Buddha	Fig. 8	242	1000×1000	0.34	0.66	10.51	0.39	11.56	30.50	2.6×	0.1920
Conference	Fig. 9a	900	1920×1080	1.49	2.63	11.53	0.77	15.65	115.51	7.4×	0.9656
Living Room	Fig. 9b	2500	1920×1080	4.36	7.84	11.30	0.88	20.02	325.07	16.2×	0.7445
Fertility	Fig. 11	800	1920×735	1.46	2.26	28.56	0.65	31.47	133.01	4.2×	0.0340
White Room	Fig. 12	722	1920×1080	1.19	2.07	14.58	1.21	17.86	119.49	6.7×	0.0779

Table 3: *Timings in ms obtained on our different scenes using a 8^3 3D grid. The rendering without interpolation is used as reference. We give the computation times for both the coefficients and the coefficients plus their gradients to evaluate the extra cost of gradient computations.*



(a) Analytical *(b)* Ours *(c)* 100× Abs. diff.

Figure 5: *Validation of our method to compute the gradients of SH basis functions (top line, Sec. 4.3) and the gradients of SH coefficients for spherical lights (bottom line, Eq.12) using $(l,m) = (6,3)$. Column (a) computed using the differentiation of the analytical equations [Slo08](Sec. 4.1) and (b) computed with our methods. We show in (c) the difference between (a) and (b) to highlight the accuracy of our methods. Spatial gradients are encoded as colors such that each pixel corresponds to a different pair (θ,ϕ), θ varies according to the columns, and ϕ varies according to the rows of each image. The gradients for the SH coefficients are computed using a spherical light at a distance of 2 and a radius equal to 0.5.*

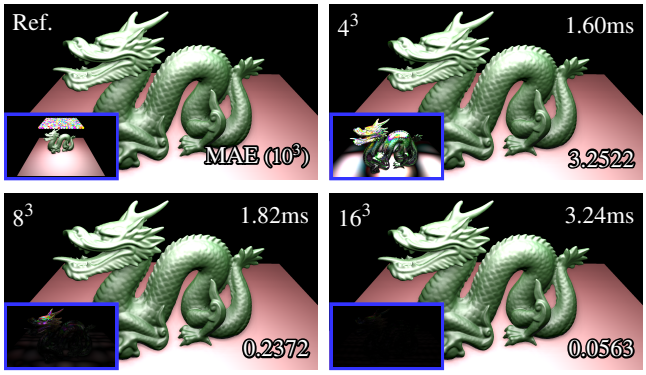


Figure 6: *Results obtained by varying the 3D grid resolution with Hermite interpolation and 578 lights. Timings at the top right of each images indicate the computation time of the SH coefficients and their gradients on the 3D grid. The reference is computed with SH without interpolation. The inset of reference image shows the scene configuration and other insets show 100× absolute errors.*

illustrate a benefit of using spherical lights when representing textured lights in Fig. 9. In this case, a spherical light is placed at each texture pixel, and the results always closely match the SH reference (Table 3).

Performance We present in Table 3 the results produced on different scene configurations. We observe that with ≈500 lights we obtain an acceleration of $\approx 5\times$ compared to the SH reference. This acceleration is even higher when the number of sources increases. On average, we obtain an overhead of $\approx 1.8\times$ to evaluate the gradient of the coefficients in addition to the computation of the coefficients themselves. This shows that, as depicted in the previous sections (Eq. 17), we are able to reuse temporary results between the computations of the coefficients and their gradients.

In our renderings, the light is projected on a full set of SH and the material is an isotropic GGX-based microfacet model projected only on ZH. The convolution between light and material is computed respectively between a function projected on SH (for the light) and another only on ZH (for the material) [SKS02]. When interpolating SH coefficients in a grid, the radiance function cannot be reduced to a ZH representation. We thus cannot take advantage of the fast convolution computation between two ZH following Mézières *et al.* [MDVP22]. Despite this, the gain provided by the grid interpolation over the direct projection of light sources pays off, and the overhead produced by the convolution SH by ZH becomes negligible as the number of lights increases (Fig. 10).

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

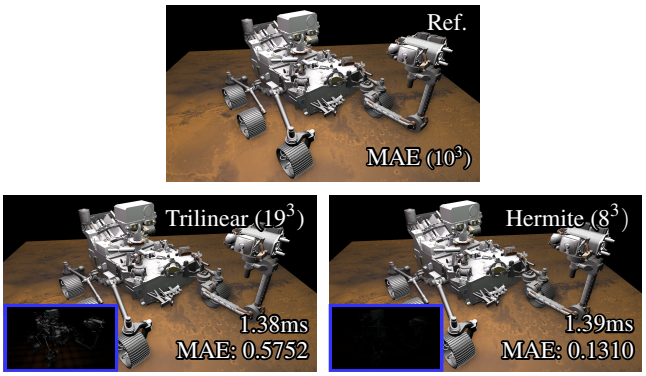
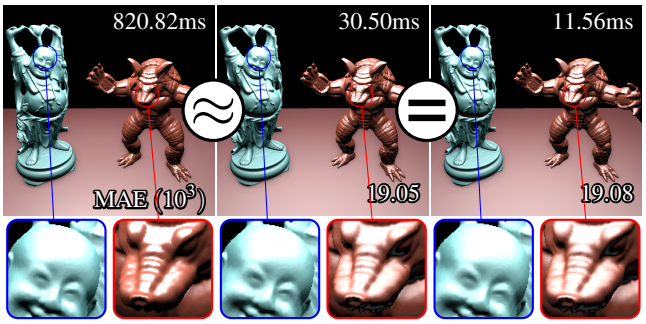


Figure 7: *Equal-time grid generation comparison using 512 lights. The same time is required to compute a 19^3 3D grid without gradient (for trilinear interpolation) (bottom-left) and a 8^3 3D grid with gradients (for Hermite interpolation) (bottom-right). When trilinearly interpolating the very large grid is still not dense enough to surpass the quality of the Hermite interpolation. Insets show $40\times$ absolute errors.*



(a) Ray casting *(b) SH reference* *(c) Ours*

Figure 8: *Comparison against ray casting with a lambertian Buddha, a glossy Armadillo and 242 lights. As demonstrated for polygonal lights [WCZR20], our approximation with interpolation (c) almost perfectly match the result without interpolation (b). Both results are close to the ray cast reference. Looking at the armadillo, results only differ on glossy material due to the order limit on SH approximation. The ray cast result is generated using 100 importance samples for each light.*

In addition, the convolution between two ZH functions has multiple inconveniences. Firstly, the cosine of the rendering equation is approximated by the cosine between the surface normal and the direction to the center of the sphere, which significantly affects the Mean Absolute Error (MAE) (Table 4). Secondly, we cannot use this method with numerical materials [KSS02] (Fig. 11) since SH coefficients for the BRDF cannot be directly reduced to ZH.

Triangular light approximation In order to evaluate our approximation of polygonal lights (Fig. 12), we propose to consider a square light source split in small triangles (as done in [WCZR20]), and then replace each triangle by our spherical approximation (Sec. 4.4). For our comparison, we use a slightly modified version of the code published by Wu *et al.* [WCZR20]. We compute the solid angle substended by a triangle using van Oosterom's formula [VS83]:

$$\tan\frac{\Omega}{2} = \frac{(\vec{u}_1 \times \vec{u}_2)\cdot\vec{u}_3}{1+\vec{u}_1\cdot\vec{u}_2+\vec{u}_2\cdot\vec{u}_3+\vec{u}_3\cdot\vec{u}_1}\,, \tag{29}$$

in which $\vec{u}_i = (p_i - O)/(\|p_i - O\|)$ is the normalized direction from the origin O to the vertex p_i of the triangle, and the index i is circular $\vec{u}_4 = \vec{u}_1$ and $\vec{u}_0 = \vec{u}_3$. The initial formula creates numerical inaccuracies when a triangular light is far away or when many polygonal lights have aligned edges. This strongly affects the comparison between triangular lights and our approximation. We analyse three main criteria of scalability: the number of light sources, the 3D grid resolution, and the SH band limit. Our approximation of triangular lights creates only a small overhead over spherical lights and it is more efficient to compute than the original triangular light (Fig. 13). We also observe a significantly better scalability of our approximation when increasing the number of SH bands (Table 2) or increasing the 3D grid resolution (Fig. 14). Our representation also provides an accurate approximation of the SH coefficients of triangular lights (Fig. 15). The MAE is reduced by one order of magnitude compared to raw spherical lights (Table 4). The approximation error of the SH coefficients has a quadratic behavior according to the number of SH bands (Fig. 15d). However, using the number of SH bands traditionally considered in real-time rendering, *i.e.* 5 to 9 SH bands, the error on the coefficients approximation remains very low, in an order of 10^{-8}, and the error in the final rendering is only slightly increased (Fig. 16).

Figure 9: *Textured lights rendering with 900 lights (a) and 2500 lights (b). The insets illustrate the scene configurations.*

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

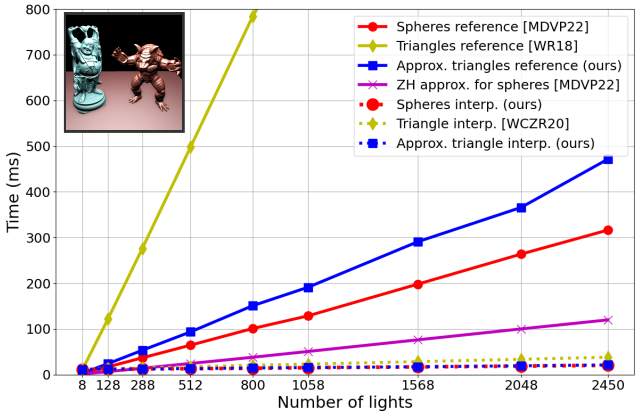


Figure 10: *Timings (in ms) to compute the final shading,* i.e. *SH coefficients and gradients plus interpolation, on a* 8^3 *3D grid using different number of lights on Fig. 8. We compare spherical lights, triangular lights and our approximation of triangular lights, with and without interpolation. References correspond to the computation without interpolation. The ZH approximation for spheres [MDVP22] cannot be used with interpolation but efficiently approximates spherical light at low cost.*

Compared method	Fig. 8			Fig. 12
	Number of lights			
	288	512	800	722
Comparison with **spheres** [MDVP22]				
Spheres (ZH approx.) [MDVP22]	1.866	1.865	1.865	1.540
Spheres interp. (Eq. 12)	0.192	0.192	0.193	0.079
Comparison with **triangles** [WR18]				
Spheres [MDVP22]	15.86	15.76	15.67	62.08
Triangles approx. (Eq. 26)	0.096	0.068	0.121	0.129
Comparison with triangles interp. [WCZR20]				
Spheres interp. (Eq. 12)	15.93	15.83	15.75	62.13
Triangles approx. interp. (Eq. 27)	0.215	0.165	0.174	0.176

Table 4: *MAE* (10^3) *with different scene configurations with an interpolation in a* 8^3 *3D grid.* ***Bold text*** *highlights rendering without interpolation. The comparisons with spheres [MDVP22] show that the interpolation for spherical sources produces a better approximation than using the ZH acceleration that approximate the cosine in the lighting equation. The comparisons with triangles [WR18, WCZR20] show that our approximation of triangles significantly reduces the MAE compared to spherical lights.*

The triangles configuration plays a very important role in the error introduced by our approximation (Fig. 17). This error lies in the difference of the shapes, sphere against triangle, projected on the unit hemisphere over a shading point. The higher this difference, the higher the approximation error. This difference is related to the solid angle subtended by the triangle, for well-shaped triangles, or to the narrowness of the triangle in the general case. The narrowness can be quantified by the inverse of the intersection area between the triangle and a co-planar disk centered at the center of gravity of the triangle having the same area than the triangle.

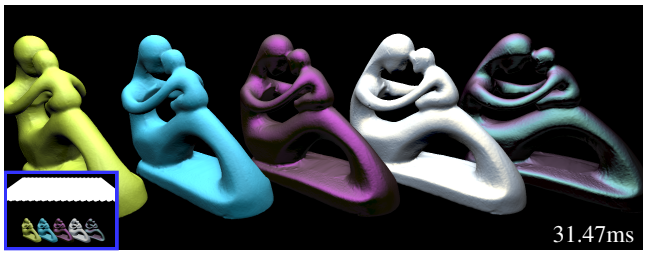


Figure 11: *Rendering using a* 8^3 *3D grid for interpolation with captured materials from [DJ18]. From left to right: cm_toxic_green, paper_blue, vch_frozen_amethyst, satin_gold, cc_nothern_aurora.*

7. Limitations and future work

As our proposal relies on Spherical Harmonics, it faces the limitations inherent to this representation, also found in the Wu *et al.* [WCZR20] approach.

Analytical and numerical integration As evoked by Wu *et al.* [WCZR20], it could be envisioned to switch between analytic and numerical integration to compute SH coefficients and their gradients. Indeed, when lights are far away from the shaded point, only one sample should be sufficient for numerical integration. Moreover, importance sampling of spherical lights is well studied in literature [DHB17]. However, in practice, sampling numerical materials (*i.e.* data-driven materials [DJ18]) on the GPU (Fig. 11) is difficult and often very expensive. Using SH thus remains a pertinent solution as SH allow the use of data-driven materials with only a very small overhead.

Textured and anisotropic lights Our method handles textured light in a very simple manner, by replacing each texel by a spherical light. However, neighboring texels in the texture may have a similar color, which introduces redundancy in the spherical light colors. This redundancy can be used to drastically reduce the number of required spherical light, e.g. by building a quadtree approximating the texture and placing a spherical light at the center of each leaf. Being orthogonal to our contributions, this is left for future work. Considering anisotropic lights, *i.e.* angularly non uniform emissive lights, the SH projection, as the gradient, should be adapted to take into account the radiance variations within the solid angle subtended by the spherical source (Fig. 2).

Shadow and visibility Our results do not integrate the visibility other than by a rough ambient occlusion approximation. Two problems must be faced to integrate the visibility in our framework: (1) projecting the dynamic visibility on SH at each frame and (2) modifying the light-field projection at each shaded point. While the former is related to efficient SH projection of arbitrary functions and is an active research area, the later is related to handling

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

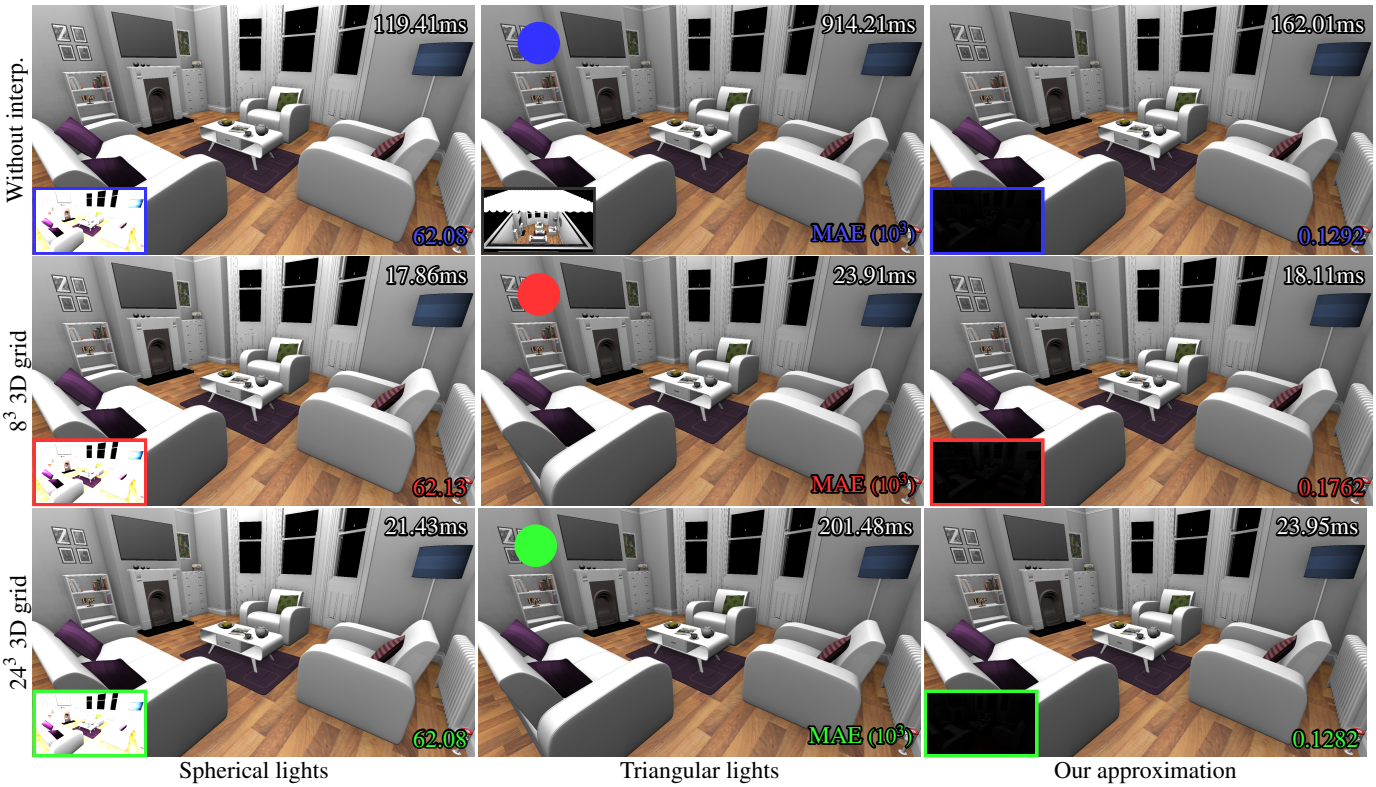


Figure 12: *Accuracy of our approximation of triangular lights [WR18, WCZR20] using interpolation. The insets show* 100× *absolute errors, insets and MAE are computed using the image with the corresponding colour patch as reference. MAE is reduced by two orders compared to raw spherical lights and the* 100× *absolute error is only slightly visible, which illustrate the accuracy of our approximation of polygonal lights. This figure also highlights that triangular lights timings increase significantly with the size of the 3D grid (Fig. 14).*

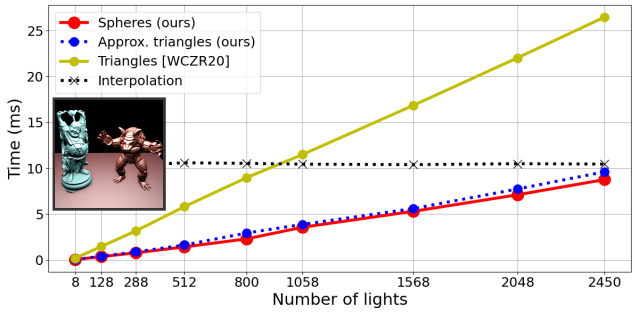


Figure 13: *Timings (in ms) to compute SH coefficients and their gradients in a* 8^3 *3D grid using different number of lights on Fig. 8. We compare spherical lights, triangular lights and our approx. of triangular lights. Interpolation does not depend on the type and number of lights.*

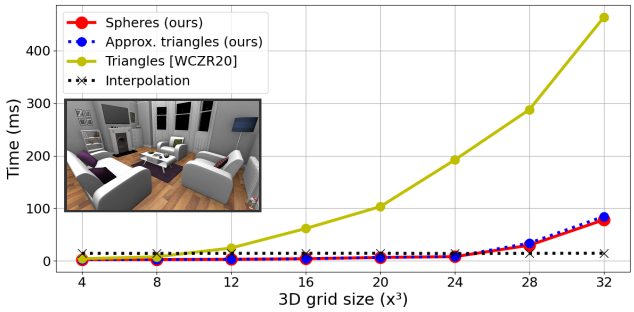


Figure 14: *Timings (in ms) to compute SH coefficients and their gradients for different 3D grid resolutions on Fig. 12 with* 722 *lights. Interpolation is not affected by the the grid resolution.*

anisotropic emission as noted above. It should also be noted that integrating visibility would add a costly general SH triple product to compute the shading, even though efficient algorithms were recently proposed [XZA*21]. Solving this visibility problem is an important future research direction while being orthogonal to our proposal that focuses on efficient gradient computation.

SH band limit for real-time rendering In [WCZR20], scalability has not been investigated when increasing the number of SH bands. The number of bands can be a bottleneck for both the SH grid computation, and the SH interpolation at shading stage. In this work, we shown that using an higher number of bands to construct the 3D grid is faster with spherical lights than with polygonal lights (Table 2). However, during interpolation, the number of memory accesses to retrieve the SH coefficients and gradients is high (espe-

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

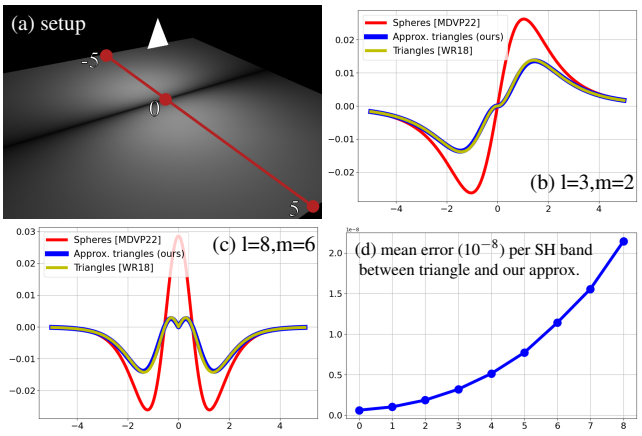


Figure 15: *Accuracy of our approximation of triangular lights SH projection [WR18]. We show the setup of the scene in (a). In (b) and (c) we evaluate one SH coefficient on the red line highlighted in (a). The plot in (d) represents the mean error (10^{-8}) between triangles and our approximation computed for each SH band on the whole plane lighted by the light.*

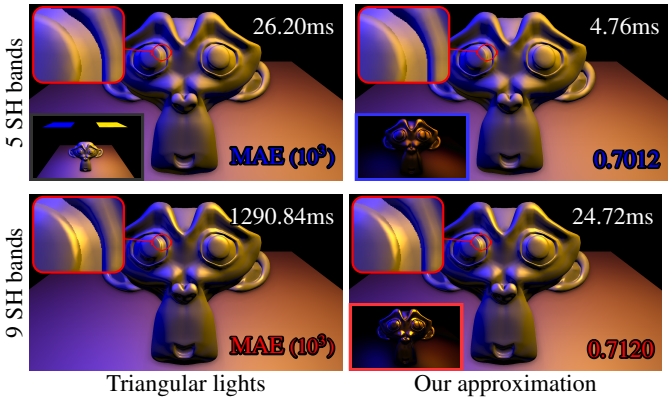


Figure 16: *The error of our approximation of triangular light [WR18] according to the number of SH bands. Results are generated with 36 lights, without interpolation and a glossy model. Bottom images inset in the left show the scene configuration and insets in the right show* 100× *absolute errors. The approximation error remains in the same magnitude order between 5 and 9 SH bands.*

cially when using a high number of SH bands) and creates a non-negligible bottleneck (Table 5). We believe that a careful low-level implementation on GPU could reduce this limitation.

Applications of SH gradients The goal of our proposal is to efficiently reconstruct the radiance field using a gradient based interpolation method. This is also the goal of Neural Radiance Field approaches (NeRF). Some recent methods use the SH and their gradients for this purpose [YLT*21, YFKT*21], hence they constitute a direct application of our method. Jimenez *et al.* [JWPJ16] propose to compute the ambient occlusion on SH using spherical kernels, *i.e.* the same framework as our proposal. Occlusion being generally smooth, this approach could take advantage of our method. However, the use of gradients in a general visibility context is a complex problem because the visibility is not generally a continuous and smooth function.

SH bands	Eval coeff & grad.	Interp.	Total	Reference	
				Time	Speed up
5	1.39	7.65	9.04	45.25	5.0×
6	1.87	8.90	10.77	53.77	5.0×
7	2.43	26.14	28.57	77.18	2.7×
8	3.07	55.57	58.64	98.79	1.7×
9	3.35	132.29	135.64	148.04	1.1×

Table 5: *Timings (in ms) with different number of SH bands using Fig. 7. Interpolation is difficult to compute in reasonable times when increasing the SH bands due to the large number of memory accesses required to retrieve the SH coefficients. Polygonal lights [WCZR20] suffer from the same problem since interpolation has also to be performed.*

Second-order derivatives Instead of only using the gradient, some existing methods go further and compute the Hessian matrix [SJJ12, MJJG18]. This matrix might be computed by deriving again the recursive relations we developed. As our method exhibits auto-similarities in its formulation, *e.g.* deriving SH basis function results in an expression using the same sub-terms but at another degree (Eq. 14, 17 and 19), this might ease this derivation. However, the mathematical and computational complexity of these derivations need to be carefully investigated and we leave this interesting direction for future investigations.

8. Conclusion

In this research, we propose a recursive analytic derivation for the gradient of SH coefficients for uniform spherical lights. Our method handles several hundred lights while offering real-time performance on scenes with dynamic geometry, lights and viewpoints. We also introduce a model to approximate the SH projection of polygonal light with spherical ones at a lowest cost and at similar quality. This approximation introduces only a small overhead compared to raw spherical lights. Our contribution also concerns the computation of the gradients of SH basis function. To the best of our knowledge, we do not know any other efficient method performing this computation and the method we propose is more efficient than the classical Cartesian method. Finally, we show that our method can be implemented in a SH framework facilitating the use of fully dynamic scenes.

Acknowledgements

This project was partially funded by the CaLiTrOp project from French National Research Agency (ANR-16-CE33-0026).

Drone taken from Sketchfab. Horse taken from Sumner and Popovic. Perseverance taken from NASA/JPL-Caltech. Dragon,

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

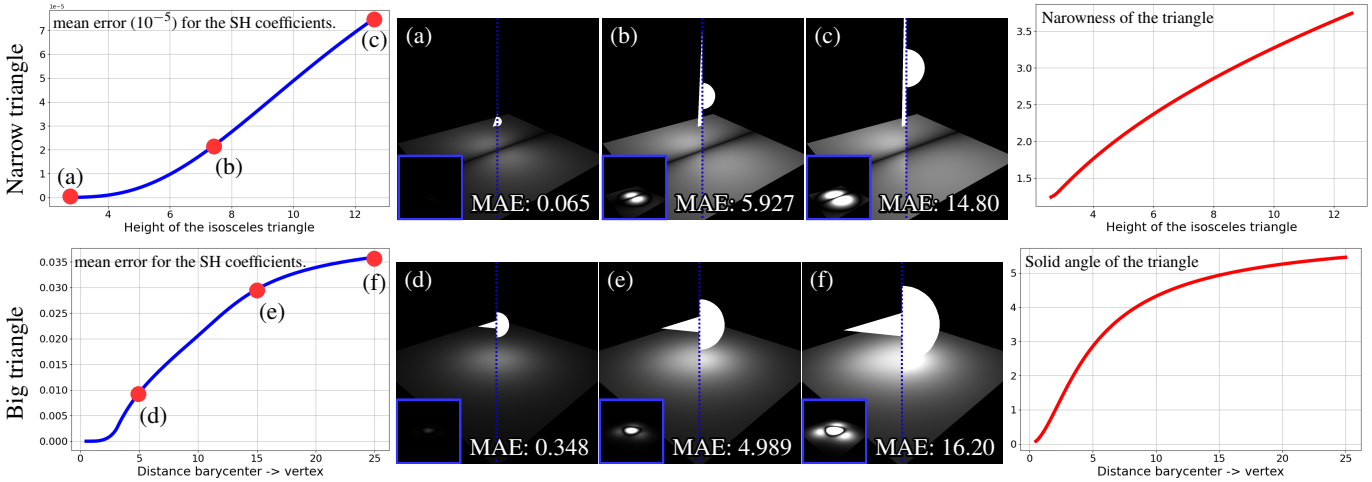


Figure 17: *Limitations and failure cases of our polygonal source approximation. Narrow triangles (first row) are known to be problematic in rendering. The approximation error is proportional to the* narrowness *of the triangle. Large triangles (second row) introduce an error related to their subtended solid angle. Insets show* 10× *absolute errors.*

Armadillo and Buddha taken from Stanford 3D scanning repository. White room and conference taken from the McGuire Computer Graphics Archive. Suzanne taken from blender foundations. Fertility taken from visionair. Living room taken from BlendSwap.

References

[AKDS04] ANNEN T., KAUTZ J., DURAND F., SEIDEL H.-P.: Spherical harmonic gradients. In *SIGGRAPH Sketches* (2004), p. 110. doi:10.1145/1186223.1186361. 3, 6

[AP21] ALLMENRÖDER J., PETERS C.: Linearly transformed spherical harmonics. *I3D poster* (2021). 2

[BS08] BAVOIL L., SAINZ M.: Screen space ambient occlusion. *NVIDIA developer information 6* (2008). 6

[BXH*18] BELCOUR L., XIE G., HERY C., MEYER M., JAROSZ W., NOWROUZEZAHRAI D.: Integrating clipped spherical harmonics expansions. *ACM Transactions on Graphics 37*, 2 (Mar. 2018). doi:10/gd52pf. 2

[CMS87] CABRAL B., MAX N., SPRINGMEYER R.: Bidirectional reflection functions from surface bump maps. In *SIGGRAPH* (1987), p. 273–281. doi:10.1145/37401.37434. 2, 3

[DHB17] DUPUY J., HEITZ E., BELCOUR L.: A spherical cap preserving parameterization for spherical distributions. *Trans. Graph. 36* (2017). doi:10.1145/3072959.3073694. 2, 10

[DJ18] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *Trans. Graph. 37*, 6 (2018). doi:10.1145/3272127.3275059. 10

[DSJN19] DUBOUCHET A., SLOAN P.-P., JAROSZ W., NOWROUZEZAHRAI D.: Impulse responses for precomputing light from volumetric media. *EG* (2019). doi:10/gf6rx8. 2

[GSHG98] GREGER G., SHIRLEY P., HUBBARD P. M., GREENBERG D. P.: The irradiance volume. *IEEE Computer Graphics and Applications 18*, 2 (1998), 32–43. doi:10.1109/38.656788. 2

[Han88] HANSEN J. E.: *Spherical near-field antenna measurements*, vol. 26. Iet, 1988. 5

[HDHN16] HEITZ E., DUPUY J., HILL S., NEUBELT D.: Real-time polygonal-light shading with linearly transformed cosines. *Trans. Graph. 35* (2016). doi:10.1145/2897824.2925895. 2

[JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *Trans. Graph. 27* (2008). doi:10.1145/1330511.1330518. 2

[JWPJ16] JIMÉNEZ J., WU X., PESCE A., JARABO A.: Practical real-time strategies for accurate indirect occlusion. *SIGGRAPH Courses* (2016). 12

[JZJ08] JAROSZ W., ZWICKER M., JENSEN H. W.: Irradiance gradients in the presence of participating media and occlusions. In *EGSR* (Goslar, DEU, 2008). doi:10.1111/j.1467-8659.2008.01246.x. 2

[Kaj86] KAJIYA J. T.: The rendering equation. In *CGIT* (1986). doi:10.1145/15922.15902. 2

[KGBP05] KŘIVÁNEK J., GAUTRON P., BOUATOUCH K., PATTANAIK S.: Improved radiance gradient computation. In *SCCG* (2005). doi:10.1145/1090122.1090148. 2

[KGPB08] KŘIVÁNEK J., GAUTRON P., PATTANAIK S., BOUATOUCH K.: Radiance caching for efficient global illumination computation. In *SIGGRAPH Classes* (2008). doi:10.1145/1401132.1401228. 2

[KGW*08] KŘIVÁNEK J., GAUTRON P., WARD G., JENSEN H. W., CHRISTENSEN P. H., TABELLION E.: Practical global illumination with irradiance caching. In *SIGGRAPH Classes* (2008). doi:10.1145/1401132.1401213. 2

[KSS02] KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *EGRW* (Goslar, DEU, 2002), p. 291–296. 9

[Leh07] LEHTINEN J.: A framework for precomputed and captured light transport. *Trans. Graph. 26* (2007). doi:10.1145/1289603.1289604. 2

[LHL*21] LYU L., HABERMANN M., LIU L., R M. B., TEWARI A., THEOBALT C.: Efficient and differentiable shadow computation for inverse problems. In *IEEE International Conference on Computer Vision (ICCV)* (2021). 3

[LTL*19] LIU H.-T. D., TAO M., LI C.-L., NOWROUZEZAHRAI D., JACOBSON A.: Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *ICLR* (2019). 3, 5

[LZBD21] LUAN F., ZHAO S., BALA K., DONG Z.: Unified Shape and SVBRDF Recovery using Differentiable Monte Carlo Rendering. *CGF 40*, 4 (2021), 101–113. doi:10.1111/cgf.14344. 3

[MDVP22] MÉZIÈRES P., DESRICHARD F., VANDERHAEGHE D.,

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.

PAULIN M.: Harmonics virtual lights : fast projection of luminance field on spherical harmonics for efficient rendering, 2022. arXiv:2201.01487. 2, 3, 4, 6, 8, 10

[MJJG18] MARCO J., JARABO A., JAROSZ W., GUTIERREZ D.: Second-order occlusion-aware volumetric radiance caching. *Trans. Graph. 37* (2018). doi:10.1145/3185225. 12

[MP21] MÉZIÈRES P., PAULIN M.: Efficient spherical harmonic shading for separable BRDF. In *SIGGRAPH ASIA 2021 Technical Communications* (Tokyo, Japan, Dec. 2021). 2, 6

[NDDJK21] NIMIER-DAVID M., DONG Z., JAKOB W., KAPLANYAN A.: Material and Lighting Reconstruction for Complex Indoor Scenes with Texture-space Differentiable Rendering. In *EGSR* (2021). doi:10.2312/sr.20211292. 3

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *Trans. Graph. 22* (2003). doi:10.1145/882262.882280. 2

[NSF12] NOWROUZEZAHRAI D., SIMARI P., FIUME E.: Sparse zonal harmonic factorization for efficient SH rotation. *Trans. Graph. 31* (2012), 1–9. doi:10.1145/2167076.2167081. 2, 6, 7

[Pet21] PETERS C.: Brdf importance sampling for polygonal lights. *Trans. Graph. 40* (2021). doi:10.1145/3450626.3459672. 2

[Ram09] RAMAMOORTHI R.: *Precomputation-based rendering*. NOW Publishers Inc, 2009. 2

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. In *SIGGRAPH* (2006). doi:10.1145/1179352.1141982. 2

[SJJ12] SCHWARZHAUPT J., JENSEN H. W., JAROSZ W.: Practical Hessian-based error control for irradiance caching. *ACM Transactions on Graphics 31*, 6 (Nov. 2012). doi:10/gbb6n4. 12

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *CGIT* (2002). doi:10.1145/566570.566612. 2, 8

[SL17] SILVENNOINEN A., LEHTINEN J.: Real-time global illumination by precomputed local reconstruction from sparse radiance probes. *SIGGRAPH Asia 36*, 6 (2017). doi:10.1145/3130800.3130852. 3

[Slo08] SLOAN P.-P.: Stupid Spherical Harmonics (SH) Tricks. *Game Dev. Conf. 9* (2008), 320–321. 4, 7, 8

[Slo13] SLOAN P.-P.: Efficient Spherical Harmonic Evaluation. *J. Comput. Graph. Tech. 2*, 2 (sep 2013), 84–90. 2, 3, 5, 6, 14

[SLS05] SLOAN P.-P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *Trans. Graph. 24* (2005), 1216–1224. doi:10.1145/1073204.1073335. 2, 4

[SS21] SILVENNOINEN A., SLOAN P.-P.: Moving Basis Decomposition for Precomputed Light Transport. *CGF* (2021). doi:10.1111/cgf.14346. 3

[VS83] VAN OOSTEROM A., STRACKEE J.: The Solid Angle of a Plane Triangle. *IEEE Transactions on Biomedical Engineering BME-30*, 2 (1983), 125–126. doi:10.1109/TBME.1983.325207. 9

[WCZR20] WU L., CAI G., ZHAO S., RAMAMOORTHI R.: Analytic spherical harmonic gradients for real-time rendering with many polygonal area lights. *Trans. Graph. 39* (2020). doi:10.1145/3386569.3392373. 2, 3, 6, 7, 9, 10, 11, 12

[WH08] WARD G. J., HECKBERT P. S.: Irradiance gradients. In *SIGGRAPH Classes* (2008). doi:10.1145/1401132.1401225. 2

[WR18] WANG J., RAMAMOORTHI R.: Analytic spherical harmonic coefficients for polygonal area lights. *Trans. Graph. 37* (2018). doi:10.1145/3197517.3201291. 2, 3, 10, 11, 12

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *SIGGRAPH* (1988). doi:10.1145/54852.378490. 2

[XZA*21] XIN H., ZHOU Z., AN D., YAN L.-Q., XU K., HU S.-M., YAU S.-T.: Fast and accurate spherical harmonics triple and multiple products. In *SIGGRAPH ASIA 2021* (Tokyo, Japan, Dec. 2021). 11

[YFKT*21] YU A., FRIDOVICH-KEIL S., TANCIK M., CHEN Q., RECHT B., KANAZAWA A.: Plenoxels: Radiance fields without neural networks, 2021. arXiv:2112.05131. 12

[YLT*21] YU A., LI R., TANCIK M., LI H., NG R., KANAZAWA A.: Plenoctrees for real-time rendering of neural radiance fields. *CoRR abs/2103.14024* (2021). arXiv:2103.14024. 12

[ZBN19] ZHAO Y., BELCOUR L., NOWROUZEZAHRAI D.: View-dependent radiance caching. In *Proceedings of Graphics Interface* (Waterloo, CAN, 2019). doi:10.20380/GI2019.22. 2

[ZMY*20] ZHANG C., MILLER B., YAN K., GKIOULEKAS I., ZHAO S.: Path-space differentiable rendering. *Trans. Graph. 39* (2020). doi:10.1145/3386569.3392383. 3

[ZWZ*19] ZHANG C., WU L., ZHENG C., GKIOULEKAS I., RAMAMOORTHI R., ZHAO S.: A differential theory of radiative transfer. *Trans. Graph. 38* (2019). doi:10.1145/3355089.3356522. 3

Appendix A: Final recurrences for SH evaluation and gradients.

We need to integrate the normalization factors of SH in the recurrences relations Q (Eq. 8) and T (Eq. 22) to obtain their final forms. Let's write:

$$J_l^m = \begin{cases} \sqrt{2}K_l^m & m > 0 \\ K_l^0 & m = 0 \end{cases}, \quad \begin{aligned} X_l^m &= J_l^m Q_l^m \\ Z_l^m &= J_l^m T_l^m \end{aligned} . \tag{30}$$

Multiplying Eq.21 by the normalization factor J_l^m, it gives us:

$$J_l^m \frac{\partial_\theta P_l^m(\cos\theta)}{\sin^m\theta} = X_l^m \frac{m\cos\theta}{\sin\theta} + Z_l^m . \tag{31}$$

Then, we extend the recurrence relations Q to integrate the normalization factors.

$$\begin{aligned} X_m^m &= J_m^m(1-2m)Q_{m-1}^{m-1} \\ X_{m+1}^m &= J_{m+1}^m(2m+1)\cos\theta Q_m^m \\ X_l^m &= \frac{(2l-1)\cos\theta X_{l-1}^m \frac{K_l^m}{K_{l-1}^m} - (l+m-1)X_{l-2}^m \frac{K_l^m}{K_{l-2}^m}}{l-m} \\ X_{m+2}^m &= J_{m+2}^m \frac{(2m+3)(2m+1)Q_m^m\cos^2\theta - (2m+1)Q_m^m}{2} \\ X_{m+3}^m &= J_{m+3}^m \frac{\cos\theta Q_m^m((2m+5)(2m+3)(2m+1)\cos^2\theta - 3(4m^2+8m+3))}{6} . \end{aligned} \tag{32}$$

These relations are the final forms developed in the code given by Sloan [Slo13]. The final recurrences for our derivatives T are then:

$$\begin{aligned} Z_m^m &= 0 \\ Z_{m+1}^m &= -J_{m+1}^m(2m+1)(\sin\theta Q_m^m) \\ Z_l^m &= \frac{(2l-1)(\cos\theta Z_{l-1}^m \frac{K_l^m}{K_{l-1}^m} - \sin\theta X_{l-1}^m \frac{K_l^m}{K_{l-1}^m}) - (l+m-1)Z_{l-2}^m \frac{K_l^m}{K_{l-2}^m}}{l-m} \\ Z_{m+2}^m &= -J_{m+2}^m(2m+3)(2m+1)(\sin\theta\cos\theta Q_m^m) \\ Z_{m+3}^m &= J_{m+3}^m \frac{(\sin\theta Q_m^m)((-(2m+5)(2m+3)(2m+1)\cos^2\theta) + (4m^2+8m+3))}{2} . \end{aligned} \tag{33}$$

© 2022 The Author(s)
Computer Graphics Forum © 2022 The Eurographics Association and John Wiley & Sons Ltd.